Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (previously presented) An in-circuit emulation system, comprising:

a microcontroller, wherein said microcontroller sends I/O read data to a virtual microcontroller, and wherein said I/O read data is followed by a conditional jump instruction that resides on said virtual microcontroller;

said virtual microcontroller coupled to the microcontroller, wherein said virtual microcontroller has means for detecting said I/O read data, and further has means for computing a speculative conditional jump address before a condition for said conditional jump instruction is satisfied after receipt of said I/O read data; and

the virtual microcontroller further having means for determining after receipt of the I/O read data from the microcontroller whether to proceed with instruction execution at a next consecutive address or at the speculative conditional jump address, wherein said virtual microcontroller executes instructions at said next consecutive address or at said speculative conditional jump address based on said means for determining such that said

microcontroller and said virtual microcontroller remain in lockstep by executing the same instruction using the same clocking signal.

2. (original) The apparatus according to claim 1, wherein the conditional jump address is computed while the I/O read data are sent from the microcontroller to the virtual microcontroller.

3. (original) The apparatus according to claim 1, wherein the microcontroller sets a zero flag if an I/O read test condition is met.

4. (original) The apparatus according to claim 3, wherein the jump condition is met if the zero flag is set.

5. (original) The apparatus according to claim 1, wherein the virtual microcontroller is implemented in a Field Programmable Gate Array.

6. (previously presented) A method of handling conditional jumps in a virtual microcontroller operating in lock-step with a microcontroller, comprising:

detecting an I/O read data sent by said microcontroller, wherein said I/O read data is followed immediately by a conditional jump instruction that resides on said virtual microcontroller;

after receipt of said I/O read data, computing a speculative conditional jump address before a condition for said conditional jump instruction is satisfied;

after receipt of the I/O read data from the microcontroller, determining whether a conditional jump condition is met; and

executing instruction based on said determination, such that said virtual microcontroller remains in lock-step execution with said microcontroller.

7. (previously presented) The method according to claim 6, wherein said executing comprises execution of next consecutive instruction in the event said conditional jump condition is not met.

8. (previously presented) The method according to claim 6, wherein said executing comprises execution of an instruction at the speculative conditional jump address in the event the conditional jump condition is met.

9. (previously presented) The method according to claim 6, wherein the conditional jump address is computed while the I/O read data are sent from the microcontroller to the virtual microcontroller.

10. (previously presented) The method according to claim 6, wherein the microcontroller sets a zero flag if an I/O read test condition is met.

11. (previously presented) The method according to claim 10, wherein the jump condition is met if the zero flag is set.

12. (previously presented) The method according to claim 6, wherein the virtual microcontroller is implemented in a Field Programmable Gate Array.

13. (previously presented) The method according to claim 6, wherein instructions are stored in an electronic storage medium for execution as program steps on a programmed processor forming a part of the virtual microcontroller.

14. (previously presented) A method of handling conditional jumps in a virtual processor operating in lock-step with a device under test, comprising:

detecting an I/O read data sent by said device under test, wherein said I/O read data is followed immediately by a conditional jump instruction that resides on said virtual processor;

after receipt of said I/O read data, computing a speculative conditional jump address before a condition for said conditional jump instruction is satisfied;

after receipt of the I/O read data from the device under test, determining whether a conditional jump condition is met; and

executing instruction based on said determination, such that said virtual processor remains in lock-step execution with said device under test.

15. (previously presented) The method according to claim 14, wherein said executing comprises execution of next consecutive instruction in the event said conditional jump condition is not met.

16. (previously presented) The method according to claim 14, wherein said executing comprises execution of an instruction at the speculative conditional jump address in the event the conditional jump condition is met.

17. (previously presented) The method according to claim 14, wherein the conditional jump address is computed while the I/O read data are sent from the device under test to the virtual processor.

18. (previously presented) The method according to claim 14, wherein the device under test sets a zero flag if an I/O read test condition is met.

19. (previously presented) The method according to claim 18, wherein the jump condition is met if the zero flag is set.

20. (previously presented) The method according to claim 14, wherein the virtual processor is implemented in a Field Programmable Gate Array.